

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6556	(sequence adj number?) or (timestamp?)	USPAT	OR	OFF	2005/02/03 13:50
L2	87455	(write adj2 read) or (read adj2 write)	USPAT	OR	OFF	2005/02/03 13:51
L3	3045	packet near3 processor\$3	USPAT	OR	OFF	2005/02/03 13:52
L4	26	restart\$3 with L1	USPAT	OR	OFF	2005/02/03 13:52
L5	2	3 and 4	USPAT	OR	OFF	2005/02/03 13:57
L6	2	(US-5878117-\$ or US-5471521-\$).did.	USPAT	OR	OFF	2005/02/03 13:56
L7	48	L1 same L2	USPAT	OR	OFF	2005/02/03 13:57
L8	1	3 and 7	USPAT	OR	OFF	2005/02/03 13:59
L9	4000	memory adj read? or memory adj write?	USPAT	OR	OFF	2005/02/03 14:00
L10	124	3 and 9	USPAT	OR	OFF	2005/02/03 14:00
L11	0	4 and 10	USPAT	OR	OFF	2005/02/03 14:00
L12	16	1 and 10	USPAT	OR	OFF	2005/02/03 14:00
L13	11950165	@ad<"20011030"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:01
L14	14	12 and 13	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:03
L15	8940	(read adj table) or (read adj buffer)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:03
L16	5001	(write adj table) or (write adj buffer)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:03
L17	246	(write adj table)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:03
L18	2400	(read adj table)	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:04
L19	59	17 and 18	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:04
L20	35	2 and 19	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:04
L21	3	1 and 20	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:04

L22	3	13 and 21	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:06
L23	0	4 and 22	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:06
L24	0	4 and 20	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:06
L25	1112	711/154.ccls.	USPAT	OR	OFF	2005/02/03 14:06
L26	414	711/158.ccls.	USPAT	OR	OFF	2005/02/03 14:07
L27	372	370/394.ccls.	USPAT	OR	OFF	2005/02/03 14:07
L28	1833	25 or 26 or 27	USPAT	OR	OFF	2005/02/03 14:07
L29	2	4 and 28	USPAT	OR	OFF	2005/02/03 14:07
L30	2	13 and 29	USPAT	OR	OFF	2005/02/03 14:07
L31	50	("6049817" "6205508" "6289238" "6393569" "5659796" "6055618" "5701495" "5887146" "6026461" "6061757" "6061757" "6101181" "6249520" "5398246" "5677918" "6195277" "6216182" "6216182" "4131950" "4321486" "4427833" "4486893" "4584697" "4627084" "4799211" "4900943" "5021942" "5224161" "5264700" "5301186" "5381408" "5399224" "5422924" "5426735" "5428616" "5461614" "5469446" "5499347" "5519693" "5523999" "5526050" "5526394" "5561670" "5608738" "5613003" "5623696" "5671226" "5689719" "5744056" "5758070").pn.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:09
L32	0	4 and 31	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:09
L33	1544	15 and 16	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:09
L34	153	9 and 33	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:10
L35	3	1 and 2 and 34	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:11
L36	0	3 and 35	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	OFF	2005/02/03 14:11



Terms used memory disambiguation or memory antialiasing and packet

Found 5,851 of 1

Sort results by

relevance



[Save results to a Binder](#)

Try an [Advanced Search](#)

Try this search in [The ACM Guide](#)

Display results

expanded form



[Search Tips](#)

☐ Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale ☐ ☐

1 [Special session on memory wall: A first glance at Kilo-instruction based multiprocessors](#)

Marco Galluzzi, Valentín Puente, Adrián Cristal, Ramón Beivide, José-Ángel Gregorio, Mateo Valero

April 2004 **Proceedings of the first conference on computing frontiers on Computing frontiers**

Full text available:  pdf(226.85 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The ever increasing gap between processor and memory speed, sometimes referred to as the *Memory Wall* problem [42], has a very negative impact on performance. This mismatch will be more severe in future processor's generation. Modern cache organizations and prefetching techniques will not be able to solve this problem. A very novel and promising technique to deal with the *Memory Wall* consists on designing processors able to maintain thousands of in-flight instructions. An example of ...

Keywords: CC-NUMA, Kilo-instruction processors, ROB, in-flight instructions, instruction window, memory wall, shared-memory multiprocessors

2 [IPStash: a Power-Efficient Memory Architecture for IP-lookup](#)

Stefanos Kaxiras, Georgios Keramidas

December 2003 **Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture**

Full text available:  pdf(293.97 KB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)



[Publisher Site](#)

High-speed routers often use commodity, fully-associative, TCAMs (Ternary Content Addressable Memories) to perform packet classification and routing (IP-lookup). We propose a memory architecture called IPStash to act as a TCAM replacement, offering at the same time, better functionality, higher performance, and significant power savings. The premise of our work is that full associativity is not necessary for IP-lookup. Rather, we show that the required associativity is simply a function of the routing table ...

3 [Cache coherence in large-scale shared-memory multiprocessors: issues and comparisons](#)

David J. Lilja

September 1993 **ACM Computing Surveys (CSUR)**, Volume 25 Issue 3



Full text available:  pdf(3.12 MB)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

5 Learning not to share

Jason Liu, David Nicol

May 2001 **Proceedings of the fifteenth workshop on Parallel and distributed simulation**

Full text available:  [pdf\(779.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
 [Publisher Site](#)

Strong reasons exist for executing a large-scale discrete-event simulation on a cluster of processor nodes (each of which may be a shared-memory multiprocessor or a uniprocessor). This is the architecture of the largest scale parallel machines, and so the largest simulation problems can only be solved this way. It is a common architecture even in less esoteric settings, and is suitable for memory-bound simulations. This paper describes our approach to porting the SSF simulation kernel to t ...

6 Architecture and implementation of a VLIW supercomputer

Robert P. Colwell, W. Eric Hall, Chandra S. Joshi, David B. Papworth, Paul K. Rodman, James E. Tornes

November 1990 **Proceedings of the 1990 ACM/IEEE conference on Supercomputing**

Full text available:  [pdf\(1.29 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Very-Long-Instruction-Word (VLIW) computers achieve high performance by exploiting the fine-grain parallelism present in sequential or vectorizable code. Multiflow's /200 and /300 VLIW systems yielded near-supercomputer performance by this means despite the relative slow (65 nS) clocks. With its much faster clock period (15 nS) and architectural improvements, the new /500 system attains approximately 4-9X the performance of its predecessors. This paper describes the /500 architecture and implem ...

7 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**


Full text available:  [pdf\(4.21 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

8 The Flux OSKit: a substrate for kernel and language research

Bryan Ford, Godmar Back, Greg Benson, Jay Lepreau, Albert Lin, Olin Shivers



October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5

Full text available:  [pdf\(2.47 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

9 EPIC compilation: Optimizations to prevent cache penalties for the Intel® Itanium® 2 Processor

Jean-Francois Collard, Daniel Lavery

March 2003 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization**

Full text available:  [pdf\(1.28 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)
[Publisher Site](#)

This paper describes scheduling optimizations in the Intel® Itanium® compiler to prevent cache penalties due to various micro-architectural effects on the Itanium 2 processor. This paper does not try to improve cache hit rates but to avoid penalties, which probably all processors have in one form or another, even in the case of cache hits. These optimization make use of sophisticated methods for disambiguation of memory references, and this paper examines the performance improvement obtained ...

10 Parallel processing: a smart compiler and a dumb machine

Joseph A. Fisher, John R. Ellis, John C. Ruttenberg, Alexandru Nicolau

June 1984 **ACM SIGPLAN Notices , Proceedings of the 1984 SIGPLAN symposium on Compiler construction**, Volume 19 Issue 6



Full text available:  [pdf\(1.05 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Multiprocessors and vector machines, the only successful parallel architectures, have coarse-grained parallelism that is hard for compilers to take advantage of. We've developed a new fine-grained parallel architecture and a compiler that together offer order-of-magnitude speedups for ordinary scientific code.

11 Analytic evaluation of shared-memory systems with ILP processors

Daniel J. Sorin, Vijay S. Pai, Sarita V. Adve, Mary K. Vernon, David A. Wood

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture**, Volume 26 Issue 3

Full text available:  [pdf\(1.45 MB\)](#)  Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)
[Publisher Site](#)

This paper develops and validates an analytical model for evaluating various types of architectural alternatives for shared-memory systems with processors that aggressively exploit instruction-level parallelism. Compared to simulation, the analytical model is many orders of magnitude faster to solve, yielding highly accurate system performance estimates in seconds. The model input parameters characterize the ability of an application to exploit instruction-level parallelism as well as the interaction ...

12 Trajectory sampling for direct traffic observation

N. G. Duffield, Matthias Grossglauser

June 2001 **IEEE/ACM Transactions on Networking (TON)**, Volume 9 Issue 3

Full text available:  [pdf\(251.55 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)


Traffic measurement is a critical component for the control and engineering of communication networks. We argue that traffic measurement should make it possible to obtain the spatial flow of traffic through the domain, i.e., the paths followed by packets between any ingress and egress point of the domain. Most resource allocation and capacity planning tasks can benefit from such information. Also, traffic measurements should be obtained without a routing model and without knowledge of network ...

Keywords: Hash functions, Internet traffic measurement, packet sampling, traffic engineering

13 A high-performance network intrusion detection system

R. Sekar, Y. Guang, S. Verma, T. Shanbhag

November 1999 **Proceedings of the 6th ACM conference on Computer and communications security**

Full text available:  [pdf\(1.04 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In this paper we present a new approach for network intrusion detection based on concise specifications that characterize normal and abnormal network packet sequences. Our specification language is geared for a robust network intrusion detection by enforcing a strict type discipline via a combination of static and dynamic type checking. Unlike most previous approaches in network intrusion detection, our approach can easily support new network protocols as information relating to the protocols ...

14 Combining hardware and software cache coherence strategies

David J. Lilja, Pen-Chung Yew

June 1991 **Proceedings of the 5th international conference on Supercomputing**

Full text available:  [pdf\(979.07 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

15 TRIPS: A polymorphous architecture for exploiting ILP, TLP, and DLP

Karthikeyan Sankaralingam, Ramadass Nagarajan, Haiming Liu, Changkyu Kim, Jaehyuk Huh

Nitya Ranganathan, Doug Burger, Stephen W. Keckler, Robert G. McDonald, Charles R. Moore

March 2004 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 1 Issue 1

Full text available:  [pdf\(832.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes the *polymorphous* TRIPS architecture that can be configured for different granularities and types of parallelism. The TRIPS architecture is the first in a class of post-RISC, dataflow-like instruction sets called explicit data-graph execution (EDGE). The EDGE ISA is coupled with hardware mechanisms that enable the processing cores and the on-chip memory system to be configured and combined in different modes for instruction, data, or thread-level parallelism. To adapt ...

Keywords: Computer architecture, configurable computing, scalable and high-performance computing

16 The effect of real data cache behavior on the performance of a microarchitecture that supports dynamic scheduling

Michael Butler, Yale Patt

September 1991 **Proceedings of the 24th annual international symposium on Microarchitecture**

Full text available:  [pdf\(691.87 KB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

17 Trajectory sampling for direct traffic observation

N. G. Duffield, M. Grossglauser

August 2000 **ACM SIGCOMM Computer Communication Review , Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication**, Volume 30 Issue 4

Full text available:  [pdf\(421.07 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Traffic measurement is a critical component for the control and engineering of communication networks. We argue that traffic measurement should make it possible to obtain the spatial flow of traffic through the domain, i.e., the paths followed by packets between any ingress and egress point of the domain. Most resource allocation and capacity planning tasks can benefit from such information. Also, traffic measurements should be obtained without a routing model and without knowledge of network ...

18 DOD standard transmission control protocol

Jon Postel

October 1980 **ACM SIGCOMM Computer Communication Review**, Volume 10 Issue 4

Full text available:  [pdf\(4.83 MB\)](#) Additional Information: [full citation](#), [references](#)

19 New models and architectures: Spatial computation

Mihai Budiu, Girish Venkataramani, Tiberiu Chelcea, Seth Copen Goldstein

October 2004 **Proceedings of the 11th international conference on Architectural support for programming languages and operating systems**

Full text available:  [pdf\(573.00 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes a computer architecture, *Spatial Computation* (SC), which is based on

the translation of high-level language programs directly into hardware structures. SC program implementations are completely distributed, with no centralized control. SC circuits are optimized for *wires* at the expense of computation units. In this paper we investigate a particular implementation of SC: ASH (Application-Specific Hardware). Under the assumption that computation is cheaper than co ...

Keywords: application-specific hardware, dataflow machine, low-power, spatial computation

20 Informatics: semantic processing: How to deal with ambiguities while parsing: EXAM-
a semantic processing system for Japanese language

Hidetosi Sirai

September 1980 **Proceedings of the 8th conference on Computational linguistics**

Full text available:  pdf(547.39 KB) Additional Information: [full citation](#), [abstract](#), [references](#)

It is difficult for a natural language understanding system (NLUS) to deal with ambiguities. There is a dilemma: an NLUS must be able to produce plausible interpretations for given sentences, avoiding the combinatorial explosion of possible interpretations. Furthermore, it is desirable for an NLUS to produce several interpretations if they are equally plausible. EXAM, the system described in this paper, is an experimental text understanding system designed to deal with ambiguities effectively an ...

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)